GLIDEINWMS

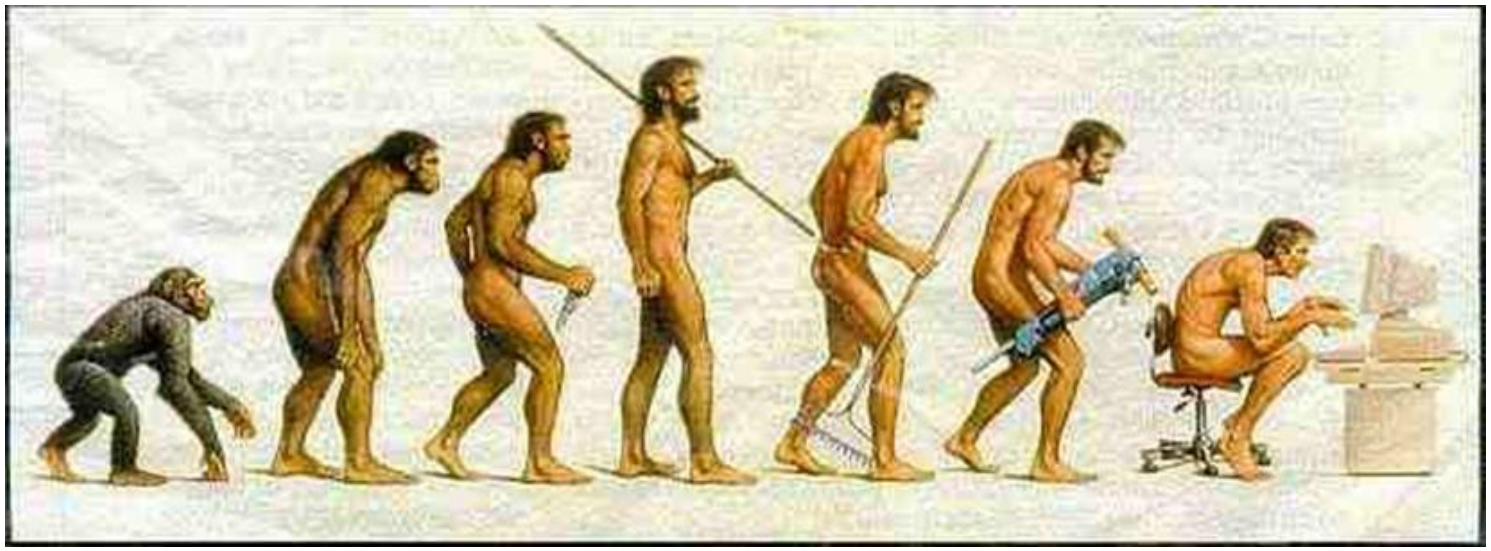
- PARAG MHASHILKAR

Department Meeting, August 07, 2013

Contents

- Evolution of Computing Grids
- Why GlideinWMS?
- GlideinWMS Architecture
- GlideinWMS & Cloud
- Demo: From a new VM to running a job successfully

Evolution of Computing Grids



No Computers ➔ Primitive Computers ➔ Super Computers/Main Frames ➔ Batch Computing

➘ ➘

Personal Computers ➔ Local Computing Clusters ➔ Computing Grids

Why GlideinWMS?

Local Computing Facilities (Accessible but LIMITED RESOURCES)	Grid Computing Facilities (WILD WILD WEST but Virtually Infinite Resources)
• Familiar setup & interface • Homogenous Resources (Condor Batch System) • In case of problems, assistance easily accessible	• Different administrative boundaries • Heterogeneous Resources • Some sites maintained well compared to others
• Local clusters maybe limited & busy when you need them	• Large number of opportunistic computing cycles available for use

Computing Clouds

- Similar to Grid Sites
- Well maintained but NOT FREE

GlideinWMS

Pilot-based WMS that creates *on demand* a *dynamically-sized overlay condor batch system* on Grid & Cloud resources to address the complex needs of VOs in running application workflows

GlideinWMS Architecture

Components

- Glidein Factory & WMS Pool
- VO Frontend
- Condor Central Manager & Scheduler

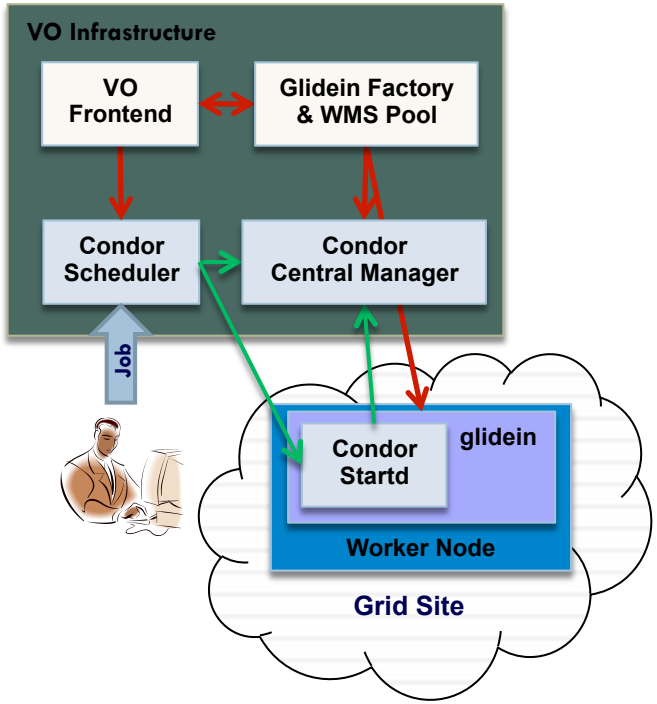
GlideinWMS in Action

- User submits a job
- VO Frontend periodically queries the condor pool and requests factory to submit glideins
- Factory looks up the requests and submits glideins to WMS Pool
- Glidein starts running on a worker node at a Grid site
- Glidein performs the required validation and on success starts condor startd
- Condor startd reports to collector
- Job runs on this resource as any other Condor batch job
- On job completion, glidein exits and relinquishes the worker node

On Demand

Dynamically Sized

Overlay Condor Batch System



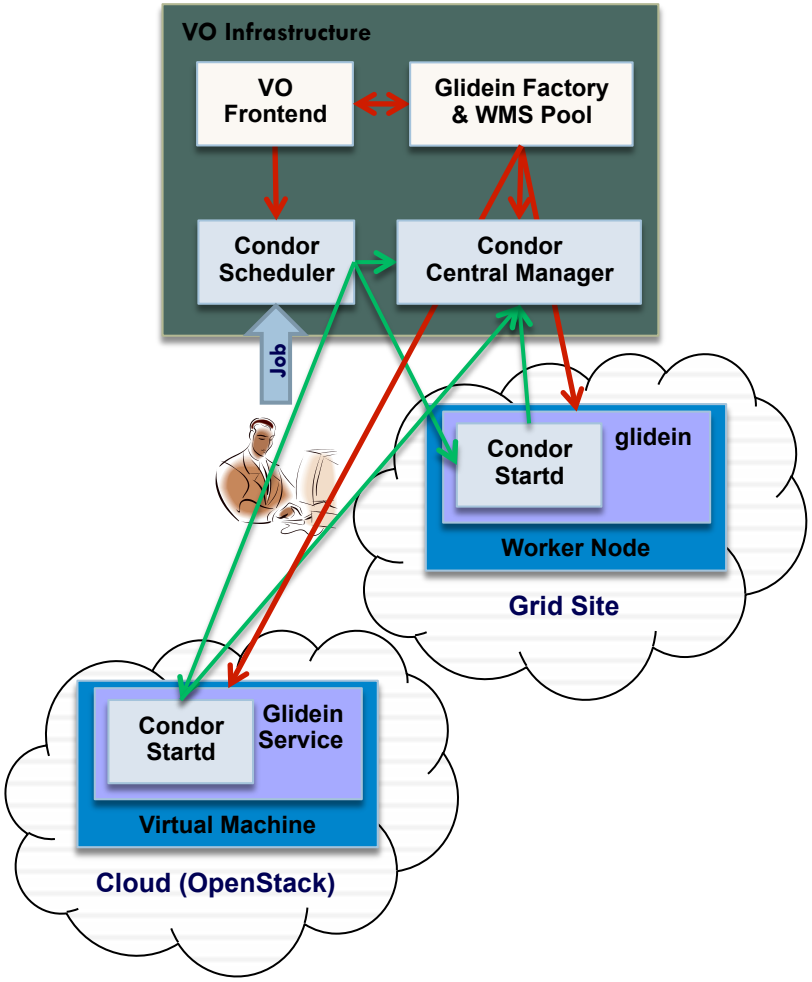
GlideinWMS: Grid Sites v/s Clouds

Grid Site

1. Factory tells the glidein how to connect back to the user pool via condor JDF
2. Glidein - CondorG job that runs on a worker node
3. Glidein shuts down after its predetermined max lifetime or after inactivity for a while.
4. Grid Site admin manages the Worker Node, i.e. software stack installed & root access to the worker node.

Cloud (EC2 Interface supported by Condor v8+)

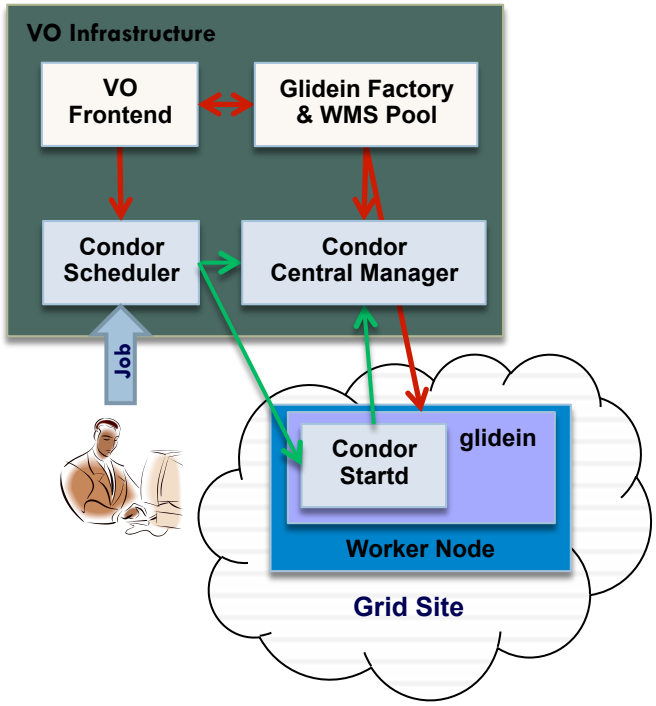
1. Factory tells the glidein how to connect back to the user pool via condor JDF
2. Glidein - Service that runs in the VM launched by condor as a job using EC2. Requires glidein rpms to be installed on the VM image.
3. Glidein shuts down after its predetermined max lifetime or after inactivity for a while. Glidein shutdown triggers VM shutdown.
4. Cloud provider facilitates hosting the VM Image provided by the VO. VO manages the image, i.e. software stack installed & root access to the worker node. Based on the policy, Cloud provider can also be the VM maintainer/administrator



Simplifying Grids & Clouds for Users

From the User's point of view –

- Users interface to Condor batch system
- Users with existing Condor-ready jobs can submit them to the grid sites and clouds with no or minimal changes
- GlideinWMS shields the user from interfacing directly with the grid sites and clouds
- Glidein validates the node before running a user job; reducing the failure rate of user jobs



From VO's point of view –

- Can prioritize jobs from different users
- Operate VO Frontend service & the Condor Pool (and optionally Glidein Factory + WMS Pool)
- Can use existing Glidein Factories operated by REX@FNAL or OSG

Users focus on Science while operations team support the operations !

Working Live Demo: Oxymoron?

- Launch a FermiCloud VM
- Setup the required user accounts
  - Factory user, Frontend user, Condor user
- Setup the required directory structure
  - HTTPD, monitoring and staging area
- Install & Configure GlideinWMS on the VM
  - Install VDT, HTTPD, m2crypto, javascriptrrd
  - Install & Configure GlideinWMS services
  - Start GlideinWMS Services
- Submit a job
- Have GlideinWMS submit glidein
- Job - Runs on the dynamically created resource